REMARKS

Claims 1, 3, 5, 9, and 10 are amended above. New claims 13 and 14 are added above. No new matter is added by the claim amendments or new claims. Entry is respectfully requested.

The drawings stand objected to under 37 CFR 1.83(a). In the drawings, FIG. 2 is amended to include the hash logic comprising an exclusive-OR operation with various inputs of claims 4 and 12, as suggested by the Examiner. Reconsideration and removal of the objections to the drawings, and entry of the amended drawings, are respectfully requested.

Claims 1, 2, 4, 5 and 10-12 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Panwar, *et al.* (U.S. Patent No. 5,890,008 - hereinafter "Panwar"). Claims 3 and 6-9 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Panwar in view of Talcott (U.S. Patent No. 6,272,623). Reconsideration and removal of these rejections, and allowance of the claims, are respectfully requested.

The present invention of amended independent claim 1 is directed to a branch predictor for a multi-processing computer able to execute multiple processes, each process having a designated process reference. A global history register stores a branch history of previous sequential branch instructions for a plurality of the multiple processes. A hash logic creates an index from a combination of a process reference of a process corresponding to a current branch instruction, an address of the current branch instruction, and the branch history for the plurality of the multiple processes. A branch prediction table stores branch prediction reference data, and outputs branch prediction reference data corresponding to the index created by the hash logic. An address selection circuit selects one of a target address known from the current branch instruction and a next address of the current branch instruction to generate a branch prediction address, in response to the branch prediction reference data output from the branch prediction table. A branch prediction result tester generating a control signal updates the branch history stored in the global history register and the branch prediction reference data stored in the branch prediction

table, in response to a comparison between a real branch address and the branch prediction address. The address selection circuit generates the branch prediction address further in response to a state of the control signal generated by the branch prediction result tester.

The present invention of amended independent claim 10 is directed to method of predicting a branch address of a conditional branch instruction with reference to a branch prediction table for storing branch prediction reference data in a multi-processing computer able to execute multiple processes, each having a designated process ID. An index is created to access the branch prediction table from a combination of a process ID of a process corresponding to the conditional branch instruction, an address of the conditional branch instruction, and a branch history comprising previous sequential branch instructions for a plurality of the multiple processes. Branch prediction reference data is read from the branch prediction table in response to the index. One of a target address known from the conditional branch instruction and a next address of the conditional branch instruction is selectively output to generate a branch prediction address in response to the branch prediction reference data, and further in response to a state of a control signal. The control signal is generated in response to a comparison between a real branch address and the branch prediction address. The branch history and the stored branch prediction reference data in the branch prediction table is updated in response to the control signal.

The present invention of amended independent claim 1 therefore includes a "global history register" that stores "a branch history of previous sequential branch instructions for a plurality of the multiple processes." In addition, a "hash logic" creates "an index" from a "combination of a process reference of a process corresponding to a current branch instruction, an address of the current branch instruction, and the branch history for the plurality of the multiple processes." This feature is illustrated, for example, at least at Fig. 2 of the present specification, which depicts a global history register (52) that stores a branch history (GH) of previous sequential branch instructions for a plurality of the multiple processes (see Fig. 2 and page 9, lines 4-6 of the present specification). In this example, a hash logic (54) creates an index (HI) from a combination of a process reference of a process corresponding to a current branch instruction

(Process ID), an address of the branch instruction (PC), and the branch history for the plurality of multiple processes (GH).

The present invention of amended independent claim 10 creates "an index to access the branch prediction table from a combination of a process ID of a process corresponding to the conditional branch instruction, an address of the conditional branch instruction, and a branch history comprising previous sequential branch instructions for a plurality of the multiple processes." The "branch history" of claim 10 is comprised of "previous sequential branch instructions for a plurality of the multiple processes."

Further, the present invention of amended independent claim 1 includes "an address selection circuit for selecting one of a target address known from the current branch instruction and a next address of the current branch instruction to generate a branch prediction address, in response to the branch prediction reference data output from the branch prediction table", and "further in response to a state of the control signal generated by the branch prediction result tester." This feature is illustrated, for example, at least at Fig. 2 of the present specification. In the example of Fig. 2, an address selection circuit (58) generates a branch prediction address (PREADDR) in response to the reference data (T/NT) when the control signal (H/M), received from the branch prediction result tester (60), is "0", corresponding to a "miss" (see page 13, lines 12-22 of the present specification). However, when the control signal (H/M) is "1", corresponding to a "hit" that results from a match between the real branch address (REALADDR) and the branch prediction address (PREADDR), the address selection circuit performs no operation (see page 13, lines 12-13 of the present specification). Similarly, the present invention of amended independent claim 10 includes "selectively outputting one of a target address known from the conditional branch instruction and a next address of the conditional branch instruction to generate a branch prediction address, in response to the branch prediction reference data, and further in response to a state of a control signal."

It is submitted that Panwar fails to teach or suggest the present invention as claimed in

amended independent claims 1 and 10. Specifically, with regard to claim 1, it is submitted that Panwar fails to teach or suggest "a global history register for storing a branch history of previous sequential branch instructions for a plurality of the multiple processes." While Panwar is cited in page 3 of the Office Action as teaching a branch predictor comprising "a history register...for storing a branch history of previous sequential branch instructions", no mention is made in Panwar as to a "global history register for storing a branch history of previous sequential branch instructions for a plurality of the multiple processes", as claimed in amended independent claim 1. Instead, Panwar discloses a plurality of virtual processors, wherein each virtual processor has a unique, independent branch history register (see Panwar, Fig. 5 and column 8, lines 42-47). In Panwar, a processor create unit (200) generates a signal that selects, at multiplexer (517), one active branch history register (BHR) at a time, in a round-robin fashion, from the virtual processors that are in an active state (see Panwar, column 8, lines 56-61). Thus, Panwar does not disclose a "global history register" that is shared and common, or "global", among the "plurality of the multiple processes", as claimed in amended independent claim 1. For similar reasons, Panwar fails to teach or suggest "a branch history comprising previous sequential branch instructions for a plurality of the multiple processes", as claimed in amended independent claim 10.

In addition, Panwar fails to teach or suggest "a hash logic for creating an index from a combination of a process ID corresponding to a current branch instruction, an address of the current branch instruction, and the branch history for the plurality of the multiple processes", as claimed. Since Panwar discloses a branch history register corresponding to each virtual processor, and not a "global history register for storing a branch history...for a plurality of the multiple processes", to identify processes in a multi-process environment, Panwar does not teach or suggest the use of a "process reference of a process", as claimed in amended independent claim 1, or a "process ID of a process", as claimed in amended independent claim 10. Thus, Panwar does not teach or suggest a "hash logic for creating an index from a combination of a process reference of a process corresponding to a current branch instruction, an address of the current branch instruction, and the branch history for the plurality of the multiple processes", as claimed in

amended independent claim 1, nor does Panwar teach or suggest "creating an index...from a combination of a process ID of a process corresponding to the conditional branch instruction, an address of the conditional branch instruction, and a branch history comprising previous sequential branch instructions for a plurality of the multiple processes", as claimed in amended independent claim 10. While each virtual process in Panwar may have a uniquely assigned branch history register (BHR_0, BHR_1, BHR_2, BHR_3), the processes are not assigned a "process reference" (claim 1) or "process ID" (claim 10) that is used to create "an index", as claimed in amended independent claims 1 and 10.

It is further submitted that Panwar fails to teach or suggest an "an address selection circuit for selecting one of a target address known from the current branch instruction and a next address of the current branch instruction to generate a branch prediction address, in response to the branch prediction reference data output from the branch prediction table" and "further in response to a state of the control signal generated by the branch prediction result tester", as claimed in amended independent claim 1. While Panwar can generate a branch prediction address, Panwar does not select "one of a target address known from the current branch instruction and a next address of the current branch instruction...in response to a state of the control signal generated by the branch prediction result tester", as stated in amended independent claim 1. Instead, Panwar generates a branch not taken (BNT) address when there is a miss (see Panwar, column 9, lines 50-57). Thus, Panwar does not generate a "branch prediction address" in response to the "branch prediction reference data", and further in response to "a state of the control signal generated by the branch prediction result tester", as claimed in amended independent claim 1. For similar reasons, Panwar fails to teach or suggest "selectively outputting one of a target address known from the conditional branch instruction and a next address of the conditional branch instruction to generate a branch prediction address, in response to the branch prediction reference data, and further in response to a state of a control signal", as claimed in amended independent claim 10.

It is therefore submitted that amended independent claims 1 and 10 are in condition for allowance, and such allowance is respectfully requested. With regard to the various dependent claims, it follows that these claims should inherit the allowability of the independent claims from which they depend.

Closing Remarks

It is submitted that all claims are in condition for allowance, and such allowance is respectfully requested. If prosecution of the application can be expedited by a telephone conference, the Examiner is invited to call the undersigned at the number given below.

Respectfully submitted,

Date: July 30, 2004

Mills & Onello, LLP
Eleven Beacon Street, Suite 605
Boston, MA 02108
Telephone: (617) 994-4900, Ext. 4902
Facsimile: (617) 742-7774

J:\SAM\0192\073004 AAF\amendmentb.wpd

Anthony P. Onello, Jr.
Registration Number 38,572
Attorney for Applicant